

PENGEMBANGAN SERVICES UNTUK MENGHUBUNGKAN APLIKASI HUMAN RESOURCE INFORMATION SYSTEM (HRIS) DENGAN MODUL HR SAP

Meyliana

Jurusan Sistem Informasi Fakultas Ilmu Komputer Universitas Bina Nusantara
Jln. K. H. Syahdan No. 9, Kemanggis/Palmerah, Jakarta Barat 11480
Telp. (021) 5345830 ext 2250, Faks. (021) 5300244
meyliana@binus.edu

ABSTRACT

Rapid business changes also trigger changes in interconnected applications. Changing multiple interrelated applications, have a level of difficulty and complexity of its own, especially if the platform and framework of these applications are not the same. One of the concepts that can help bridge the time gap problem between business change with changes in applications and platforms and frameworks to bridge the gap between applications is the implementation of "services". By using "services", interconnected applications will be able to produce more realtime information. A case study for the development of "service" is performed at the Human Resource (HR) department, which uses two different applications of Human Resource Information System (HRIS) for the frontend and use the SAP HR module as its backend. Bottom-up strategy is chosen as a strategy in developing the "service". Application of "service" is expected to help accelerate change in the development of applications and facilitate the further application, and can result in real-time information.

Keywords: Service, HRIS, Modul HR SAP, Strategi Bottom Up

ABSTRAK

Perubahan bisnis yang sangat cepat memicu perubahan pada beberapa aplikasi yang saling berhubungan. Mengubah beberapa aplikasi yang saling terkait, memiliki tingkat kesulitan dan kerumitan tersendiri, apalagi jika platform dan framework dari aplikasi-aplikasi ini tidak sama. Salah satu konsep yang dapat membantu menjembatani masalah gap waktu antara perubahan bisnis dengan perubahan aplikasi serta menjembatani gap platform dan framework antar aplikasi adalah dengan penerapan "services". Dengan menggunakan "services", aplikasi saling berhubungan dengan lebih mudah dan menghasilkan informasi yang lebih realtime. Studi kasus untuk pengembangan "service" ini dilakukan pada bagian Human Resource (HR) yang menggunakan dua aplikasi yang berbeda yaitu Human Resource Information System (HRIS) untuk frontend-nya dan menggunakan modul HR SAP sebagai backend-nya. Strategi yang dipilih dalam mengembangkan "service" ini adalah strategi "bottom up". Penerapan "service" ini diharapkan dapat membantu mempercepat perubahan aplikasi dan memudahkan dalam pengembangan aplikasi selanjutnya, serta dapat menghasilkan informasi yang real time.

Kata kunci: Service, HRIS, Modul HR SAP, Strategi Bottom Up

PENDAHULUAN

Persaingan bisnis yang sangat ketat membuat perusahaan berlomba untuk melayani pelanggannya dengan lebih baik dan lebih *customer-oriented*. Teknologi informasi menjadi salah satu faktor yang paling berpengaruh dalam menyiapkan layanan perusahaan kepada pelanggannya. Kita mengetahui bagaimana teknologi informasi dapat membantu mempermudah aktivitas dan membuat hidup kita menjadi lebih simpel. Oleh karena itu, dalam mengubah bisnis strateginya, perusahaan pasti meminta perubahan pada aplikasi yang mendukung strategi bisnis tersebut. Mengubah aplikasi yang saling terkait memiliki tingkat kerumitan dan kesulitan yang tinggi, apalagi jika *platform* atau *framework*-nya juga berbeda. Konsep pemrograman yang umum dilakukan saat ini (*hard coding*) membuat tingkat kesulitan dalam mengubah aplikasi menjadi semakin tinggi sehingga waktu yang dibutuhkan pun semakin lama. Selain itu, hal ini juga membuat informasi yang dihasilkan tidak selalu bisa *real time* diterima oleh pengguna. Penerapan aplikasi yang tepat dapat memberikan keuntungan bagi perusahaan dan dapat mendukung kinerja yang lebih baik. Kecepatan dan ketepatan data ataupun informasi yang diperoleh menjadi bagian penting yang harus diperhatikan dalam penerapan suatu aplikasi. Selain memperhatikan ketepatan dan kecepatan data ataupun informasi yang diperoleh, suatu aplikasi juga harus terintegrasi untuk dapat mendukung suatu kinerja agar dapat berjalan dengan lebih baik. Dengan memiliki aplikasi yang terintegrasi maka data maupun informasi dapat diberikan secara *real time*, sehingga suatu proses perolehan informasi dapat berjalan dengan lebih efektif dan efisien (Dabhi, Prajapati, Dashi, & Chokshi, 2009). Agar kinerja suatu perusahaan dapat berjalan dengan baik tentunya harus didukung dengan kinerja karyawan yang baik juga. Oleh karena itu diperlukan suatu aplikasi yang dapat memantau dan mengendalikan kinerja karyawan.

Penelitian ini dilakukan pada Bagian Human Resource (HR) suatu Perguruan Tinggi yang menggunakan 2 (dua) aplikasi berbeda untuk aplikasi frontend dan aplikasi backend-nya. Aplikasi frontend menggunakan web portal (disebut dengan Human Resource Information System/HRIS) yang dibangun menggunakan bahasa pemrograman MS .NET (C#), sedangkan aplikasi backend menggunakan aplikasi SAP untuk modul HR. Penelitian ini bertujuan untuk mendapatkan *services* yang dapat menghubungkan aplikasi HRIS dengan aplikasi SAP dan membangun *services* yang bersifat *reusable*, *autonomy* dan *loose coupling*. Manfaat dari penelitian ini agar data atau informasi dapat diakses dan diproses secara *real time* sehingga kinerja karyawan menjadi lebih efektif dan efisien, serta memudahkan pengembangan aplikasi selanjutnya dengan menggunakan *web services* yang sudah ada (Jarir, Quafafou, & Erradi, 2010).

METODE

Penelitian ini dilakukan dalam bentuk menganalisa sistem lama dan mendesain solusi baru (menggunakan *services*) untuk sistem yang sama. Pada sistem yang lama, hubungan antara HRIS dengan Modul HR SAP menggunakan file template sehingga berdasarkan analisa terdapat beberapa kelemahan seperti data atau informasi tidak *real time*, adanya kesalahan sehabis mati listrik karena akses harus manual, adanya kehilangan data karena tidak berhasil mendapatkan koneksi data baru, dll. Beberapa teori yang mendukung solusi yang ditawarkan akan dibahas pada subbagian di bawah ini.

Modul HR SAP

SAP (*System Application and Product*) merupakan sistem informasi yang terintegrasi, yang menyediakan informasi terintegrasi dari bagian Akuntansi sampai bagian Manufaktur, dan dari

Penjualan ke Pelayanan. Sistem SAP mendukung dan mengintegrasikan ribuan proses bisnis dengan menggunakan database tunggal.

SAP terdiri dari beberapa modul, yaitu: *Logistic*, *Accounting*, dan *Human Resource*. Modul *Human Resource* (HR) mendukung fungsi-fungsi seperti perekrutan, penggajian, pelatihan, penentuan kompensasi, dan lain sebagainya. HR terdiri dari modul antara lain:

- a. *Organizational Management* (OM), OM mendeskripsikan bermacam-macam unit bisnis di dalam organisasi, hubungan diantara bisnis dan membuat struktur organisasi. OM dapat dibagi berdasarkan fungsional atau kriteria, sebagai contoh:
 - *Jobs*, merupakan deskripsi umum atau template yang diterapkan untuk beberapa posisi dengan persyaratan tertentu.
 - *Positions*, merupakan unit organisasi terkecil. *Positions* menggambarkan tugas-tugas para individu di dalam organisasi.
 - *Persons*, merupakan orang yang kita tugaskan untuk pekerjaan tertentu.
 - *Tasks*, merupakan deskripsi dari Job
- b. *Personnel Management*, integrasi antara perekrutan dan *Personnel Administration* memungkinkan data pelamar dapat ditransfer sebagai data karyawan. Ketika data pelamar yang telah dibandingkan sesuai dengan persyaratan, maka sistem akan mentransfer data pelamar tersebut ke dalam master karyawan.
- c. *Personnel Development*, pada bagian ini perusahaan dapat merencanakan dan membuat pengembangan *personnel* dan pendidikan lebih lanjut serta pelatihan karyawan, dengan mengintegrasikan dengan *Training* dan *Event Management*. *Personnel Development Requirement* merupakan hasil dari perbandingan kebutuhan/persyaratan dari sebuah pekerjaan dengan kualifikasi dari karyawan. *Qualifying Actions* (seperti program pelatihan atau kursus) dapat diberikan kepada karyawan menggunakan *Development Plans*.
- d. *Time Management*, mendukung semua proses yang berhubungan dengan perencanaan, pencatatan, evaluasi, kehadiran dan absen karyawan.
- e. *Payroll*, berhubungan dengan kalkulasi masing-masing gaji karyawan untuk perkerjaannya yang telah dihasilkan. *Payroll* terdiri dari beberapa proses yaitu pembuatan hasil penggajian, laporan gaji, transfer bank dan pembayaran gaji.

Web Services

Menurut pendapat Tatnall (2005), *web service* adalah program, yang menerima dan menanggapi permintaan melalui internet. Secara khas, *web service* menerima permintaan dalam format berbasis XML. Menurut Cardoso (2007), *web service* didasarkan pada teknologi komputer terdistribusi dan menyediakan jenis standar pengoperasi antar perangkat lunak aplikasi berbeda dan dalam batasan organisasi menggunakan protokol dan format XML. Format aktual permintaan dan tanggapan bergantung pada standar XML yang digunakan. Salah satu standarnya adalah SOAP. Menurut Erl (2005), *Service Oriented Architecture* (SOA) adalah istilah yang mewakili model yang mana logika automasi diuraikan ke unit logika yang lebih kecil dan berbeda. Secara kolektif, unit ini meliputi bagian logika automasi bisnis yang lebih besar. Secara individu, unit ini dapat didistribusikan. *Web Service* adalah aplikasi software yang memiliki kegunaan melalui web protokol; yang terbuka dan terstandarisasi untuk aplikasi/pengguna yang lain. *Web Service* ini memperbolehkan aplikasi-aplikasi saling menjalankan dalam lingkungan yang tidak terikat dengan kuat (Dabhi, Prajapati, Dashi, & Chokshi, 2009).

Remote Function Calls (RFC)

Menurut Signorile (2007), *Remote Function Calls* (RFC) adalah tampilan pemrograman standard untuk membuat remote calls antara program-program yang terletak pada sistem yang berbeda. RFC adalah *interface* logika bertingkat lebih tinggi dari CPIC, dan dapat memudahkan *programmer* karena mereka tidak perlu khawatir mengenai implementasi rutinitas komunikasi. Dengan

interface RFC, panggilan fungsi dapat dicapai antara dua sistem SAP atau antara sistem SAP dengan perangkat eksternal (seperti aplikasi Microsoft Windows). Fungsi pustaka melingkupi RFC yang mendukung bahasa pemrograman Visual Basic dan C.

Interface RFC pada dasarnya disertai dua layanan, yaitu :

- a. Interface pemanggil untuk program ABAP. Setiap program ABAP dapat memanggil fungsi khusus menggunakan kalimat pemrograman standar.
- b. Suatu API (*Application Program Interface*) RFC untuk program selain ABAP (non – SAP). SAP bahkan menyediakan generator program RFC untuk membantu mengimplementasikan program mitra RFC dalam sistem eksternal. Dengan panggilan API ini, program eksternal dapat memanggil modul fungsi ABAP dalam sistem SAP R/3. Pada waktu yang sama, dengan API RFC, program ABAP dapat menggunakan fungsi – fungsi yang disediakan program eksternal.

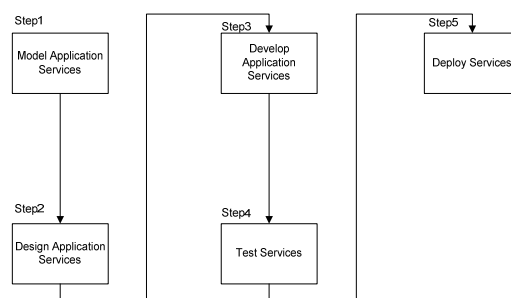
Bantuan online SAP dan file cetak dokumentasi meliputi informasi ekstensif mengenai pemrograman khusus dengan RFC dan API RFC.

Business Application Program Interfaces (BAPI)

Menurut Hernandez, Keogh, Martinez (2005), *Business Application Program Interfaces (BAPIs)* dapat digunakan sebagai mekanisme untuk mengkomunikasikan R/3 dengan aplikasi eksternal menggunakan internet. BAPI merupakan definisi entitas bisnis berorientasi objek, konsep di belakang BAPI adalah kunci dalam arsitektur kerangka bisnis dan keseluruhan strategi internet dan e-commerce SAP R/3 selama interface berorientasi objek untuk menyatukan aplikasi eksternal. Berdasarkan objek bisnis, seperti perusahaan, vendor, karyawan, material dan lain sebagainya, metode BAPI mendefinisikan metode yang dapat digunakan untuk berinteraksi dan berkomunikasi dengan objek – objek tersebut. Release 3.1 meliputi lebih dari 100 BAPI yang siap untuk menyatukan R/3 dengan solusi dan aplikasi pihak ketiga.

Strategi Bottom Up

Menurut Erl (2005), pendekatan ini pada intinya mendorong penciptaan *service* sebagai bentuk pemenuhan kebutuhan berdasar aplikasi. *Web service* dibangun pada basis yang diperlukan dan digambarkan untuk membungkus logika aplikasi untuk mendukung kebutuhan cepat akan solusi. Integrasi adalah pendukung utama bagi rancangan *bottom-up*, dimana kebutuhan untuk mengambil keuntungan dari kerangka komunikasi terbuka SOAP dapat dipenuhi dengan menyertakan *service* sebagai pelapis sistem legal.



Gambar 1 Model *Bottom Up*

Langkah 1: Menggambarkan *service* aplikasi yang dibutuhkan

Langkah ini menghasilkan definisi kebutuhan aplikasi yang dapat dipenuhi melalui penggunaan *web service*. Kebutuhan khusus ini meliputi kebutuhan untuk membangun suatu saluran

integrasi *point-to-point* antara sistem legal atau solusi B2B. Kebutuhan umum lainnya muncul dari keinginan untuk menggantikan teknologi komunikasi tradisional dengan kerangka komunikasi pesan SOAP.

Untuk solusi yang menjalankan strategi *bottom-up* untuk menyampaikan solusi berbasis *service* tingkat tinggi, *service* aplikasi juga akan digambarkan guna memasukkan logika dan peraturan bisnis yang spesifik. Dalam kasus ini, sangat mungkin jika dua lapisan *service* aplikasi akan muncul yang terdiri dari *service* turunan dan utilitas. *Service – service* tersebut yang digolongkan *reusable* dapat berlaku sebagai poin akhir aplikasi generik untuk tujuan integrasi atau mereka dapat dibentuk oleh *service* induknya.

Langkah 2 : Rancang *service* aplikasi yang dibutuhkan

Beberapa *service* aplikasi yang digambarkan dalam langkah 1 dapat disampaikan oleh *service* pelapis pihak ketiga penyewaan atau mungkin melalui penciptaan *service* proxy yang terbentuk secara otomatis. *Service* ini dapat menyediakan sedikit kesempatan bagi rancangan tambahan. *Service* aplikasi yang disesuaikan ini akan perlu menjalankan proses rancangan di mana standar rancangan akan diaplikasikan untuk memastikan konsistensi.

Langkah 3 : Membangun *service* aplikasi yang dibutuhkan

Service aplikasi dibangun menurut deskripsi *service* masing – masing dan spesifikasi rancangan yang dapat diaplikasikan.

Langkah 4 : Uji coba *service*

Service, lingkungan solusi yang terkait dengannya dan logika legal dasar diuji untuk memastikan kalau kebutuhan pemrosesan dapat dipenuhi. Kinerja dan ukuran uji tekanan sering digunakan untuk memasang parameter pemrosesan sistem legal yang dipaparkan melalui *service* pelapis. Uji keamanan juga merupakan bagian penting dari langkah ini.

Langkah 5 : Menjalankan *service*

Solusi dan *service* aplikasi dijalankan ke produksi. Pertimbangan implementasi untuk *service* aplikasi seringkali meliputi kebutuhan kinerja dan keamanan.

HASIL DAN PEMBAHASAN

Saat ini aplikasi *frontend* yang digunakan untuk mendukung kinerja para karyawan adalah menggunakan aplikasi HRIS (*Human Resource Information System*). Dengan aplikasi ini para karyawan dapat melakukan absensi, mengajukan cuti, lembur maupun pengaksesan data personal dengan mudah. Sedangkan aplikasi *backend*-nya menggunakan aplikasi SAP yang melingkupi lima modul utama, yaitu : *Organizational management (OM)*, *Personnel Management*, *Personnal Development*, *Time Management*, *Payroll*.

Berikut penjelasan tiap aktivitas yang dilakukan para karyawan di aplikasi HRIS.

a. Absensi

Pada saat karyawan hendak melakukan absen baik absen masuk maupun absen pulang, ia dapat mengaksesnya melalui menu home pada HRIS ataupun menu *absence* yang tersedia di sebelah

kanan tampilan HRIS kemudian jam masuk ataupun jam pulang tersebut akan tercatat dalam SQL.

b. Pengajuan cuti

Pada saat karyawan hendak mengajukan izin cuti maka karyawan tersebut harus melewati serangkaian prosedur proses yang telah ditetapkan. Proses itu meliputi, karyawan yang hendak mengambil cuti dapat terlebih dahulu mengakses HRIS untuk mengajukan cuti secara *online* dan sistem akan langsung memeriksa batas izin cuti yang tersedia apakah masih memungkinkan atau tidak, jika tidak maka izin cuti akan langsung ditolak namun jika masih memungkinkan maka karyawan akan melalui proses berikutnya yaitu pengecekan tenggat waktu sebelum cuti.

Pengajuan izin cuti memerlukan sebuah persetujuan dari *approver* berwenang yang bersangkutan. Jadi setelah izin cuti didaftarkan di sistem, data izin tersebut akan langsung masuk ke SQL dengan status *pending* dan menunggu persetujuan dari *approver*. Setelah tersimpan dalam SQL, sistem akan menyampaikan izin tersebut kepada *approver*, apabila *approver* telah menyetujuinya maka sistem akan secara otomatis memperbaharui status izin di SQL tersebut.

Namun, apabila cuti diajukan kurang dari seminggu sebelum hari yang bersangkutan maka izin cuti tersebut memerlukan dua buah persetujuan dari *approver* yang berwenang. Maka setelah mendapat persetujuan dari *approver*, sistem akan mengirim izin cuti tersebut kepada *approver* kedua. Setelah *approver* kedua menyetujuinya, sistem pun akan secara otomatis memperbaharui status izin dan sistem melakukan *update* terhadap batas cuti karyawan serta mencatat histori pengambilan cuti karyawan tersebut.

c. Pengajuan lembur

Pada saat karyawan hendak merencanakan untuk mengambil lembur maka ia pun dapat mengakses HRIS terlebih dahulu untuk mendaftarkan izin lemburnya secara *online*. Setelah itu, sistem akan mencatat izin lembur tersebut ke dalam SQL Server, baik izin lembur yang diambil sepanjang kurang dari sepuluh jam lebih maka karyawan membutuhkan persetujuan dari bagian *Talent Management* ataupun *approver*, namun jika izin lembur yang hendak diambil ternyata lebih dari sepuluh jam maka karyawan pun membutuhkan tiga buah persetujuan dari bagian *Talent Management*, *approver* pertama dan kedua. Dengan demikian maka sistem akan mengirimkan izin lembur tersebut kepada *approver* yang terlebih dahulu akan memeriksa histori lembur dari karyawan tersebut dan mempertimbangkan kepentingan dari lembur yang hendak diambil apakah butuh atau tidak sebelum akhirnya memutuskan. Apabila izin lembur tersebut tidak disetujui maka karyawan tidak diperkenankan untuk menjalankan lemburnya namun jika izin lembur tersebut telah disetujui oleh bagian *Talent management* ataupun *approver* maka sistem akan memperbaharui status izin lembur karyawan itu dan karyawan berhak untuk menjalankan lemburnya, sistem secara otomatis akan langsung mencatat data lembur tersebut ke dalam histori lembur dari karyawan yang bersangkutan.

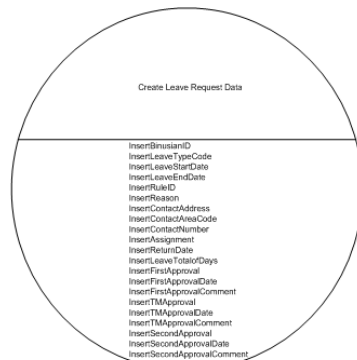
d. Pengajuan profile update

Pada saat karyawan hendak melakukan profile update terhadap data karyawan tersebut maka ia juga dapat melakukannya dengan terlebih dahulu mengakses HRIS untuk kemudian mengajukan izin perubahan data profil tersebut secara online. Setelah itu, sistem pun akan mendaftarkan izin ke dalam SQL Server dan karena pengajuan izin perubahan data ini tentunya harus dengan sepengetahuan dari seorang *approver* berwenang yang juga perlu memastikan dan menjaga keakuratan data-data karyawan yang ada dengan mempertimbangkan apakah perubahan tersebut dibutuhkan atau tidak maka sistem pun akan memberitahu kepada *approver* yang berwenang tersebut untuk meminta persetujuannya, kemudian *approval* selanjutnya diberikan oleh bagian *Talent Management*. Jika izin perubahan data profil telah disetujui maka sistem akan memperbaharui status izin profile update tersebut dan karyawan bersangkutan diperkenankan untuk melakukan profile update yang akan secara otomatis pula tercatat dalam histori karyawan tersebut.

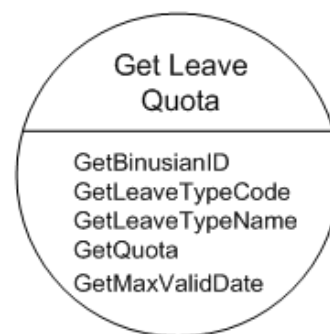
Data transaksi-transaksi yang telah tersimpan di SQL dan sudah disetujui tadi nantinya akan dikonversi ke *textfile* pada pukul 2 dini hari setiap harinya untuk kemudian dikonversikan ke SAP. Setelah pembaharuan data pada SAP selesai maka data baru tersebut akan kembali dikonversikan ke *textfile* untuk kemudian memperbaharui data di SQL Server. Sehingga dalam hal ini SQL Server bisa dianggap berperan sebagai penyimpanan data sementara sebelum akhirnya dikonversikan ke dalam SAP. Namun aplikasi HRIS ini belum terintegrasi dengan aplikasi SAP, sehingga memungkinkan perolehan data yang tidak *real time* serta adanya kehilangan data. Contohnya data yang disimpan dalam *database* (SQL) ditampung sementara pada *text file* dan kemudian pada jam yang telah ditentukan akan dipindahkan ke aplikasi SAP, namun dalam proses pemindahan data ini ada kemungkinan terjadi kehilangan data, salah satunya jika terjadi masalah pada jaringan ataupun adanya pemadaman listrik, karena pada proses ini data dalam *database* (SQL) akan terhapus secara otomatis. Melihat masalah yang ada dalam aplikasi HRIS, dibuat beberapa pengembangan pada aplikasi tersebut. Pengembangan yang dilakukan yaitu mengintegrasikan aplikasi HRIS dengan aplikasi SAP dengan arsitektur SOA yang berbasis *web service* sehingga data yang dibutuhkan dapat diakses dengan cepat dan memudahkan dalam pengembangan sistem selanjutnya karena *services* yang dibangun bersifat *reusable*. Adapun strategi yang akan dikembangkan untuk membangun service berdasarkan konsep SOA yaitu strategi *bottom – up* dengan langkah-langkah sebagai berikut :

Langkah 1: Menggambarkan service aplikasi yang dibutuhkan

Melihat dari *user interface* HRIS yang sesuai dengan proses bisnis saat ini, maka dapat ditentukan *service candidates* untuk selanjutnya dievaluasi menjadi service. Proses penentuan *service candidates* ini dapat merujuk kepada atribut dari masing-masing *user interface* HRIS.



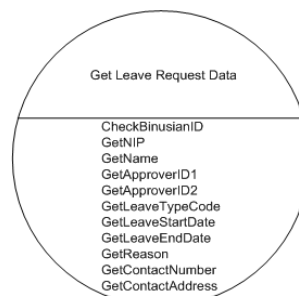
Gambar 2. Candidate Service User Interface Create Leave Request



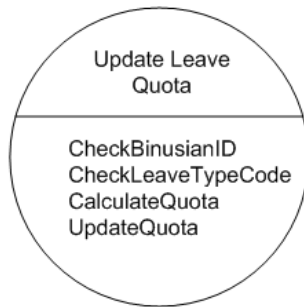
Gambar 3. Candidate Service Get Leave Quota



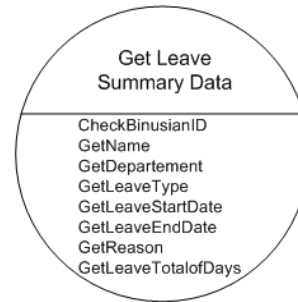
Gambar 4. Candidate Service Insert Leave Approval Data



Gambar 5. Candidate Service Get Leave Request Data



Gambar 6. Candidate Service Update Leave Quota



Gambar 7. Candidate Service Get Leave Summary

Langkah 2 : Rancang service aplikasi yang dibutuhkan

Service candidates yang telah ditentukan pada langkah 1 kemudian akan dicocokkan dengan *user interface* HRIS yang baru diusulkan dan dilihat apakah *service candidates* tersebut masih relevan. Apabila dibutuhkan, *service candidates* tadi dapat direvisi ataupun dikembangkan.

Langkah 3 : Membangun service aplikasi yang dibutuhkan

Setelah *service* selesai dirancang, proses selanjutnya akan melibatkan penerjemahan ke dalam bahasa pemrograman (*coding*) dengan menggunakan ASP.Net.

Langkah 4 : Uji coba service

Setelah proses pemrograman selesai dilakukan maka tahap berikutnya melingkupi proses testing atas *service* yang telah dibangun. Kemudian akan diperiksa apakah *service* tersebut berfungsi sesuai kebutuhan user, jika masih terdapat kekeliruan maka *service* yang telah dibangun tersebut harus direvisi kembali hingga sesuai dengan apa yang dibutuhkan.

Langkah 5 : Menjalankan service

Apabila *service* telah lolos uji coba pada tahap 4 maka *service* tersebut akan dijalankan pada platform HRIS yang baru dengan memberikan paramater guna menampilkan *output* yang diinginkan.

Aplikasi HRIS akan terintegrasi dengan SAP melalui *web service* yang berbasis arsitektur SOA. Aplikasi HRIS ini tidak perlu lagi melakukan konversi ke dalam *textfile* sebagai penampung data sementara untuk kemudian memperbaharui data pada SAP sehingga data pengajuan dan persetujuan cuti maupun lembur dapat diakses secara *real time*. Dengan akses data yang *real time* ini maka resiko kehilangan data dapat diminimalisasikan dan tentunya mendukung kinerja yang lebih efektif. Melalui pembangunan *service*, pengembangan sistem berikutnya akan menjadi lebih mudah karena *service* bersifat *reusable*. Teknologi yang digunakan dalam pengembangan aplikasi adalah *software* Visual Studio .Net 2008 *Framework* 3.5, RFC Connector dan Microsoft SQL Server 2000 sebagai basis data. Bahasa pemrograman yang digunakan antara lain ASP.Net, C#, HTML, CSS, dan XML. Perubahan aplikasi ini memicu perubahan pada proses bisnis, yaitu:

a. Absence (Absensi)

Absen kehadiran maupun kepulangan karyawan dapat dilakukan pada saat karyawan tersebut telah melakukan *login* dan masuk ke tampilan *home* dari *account* HRIS miliknya ataupun membuka tampilan khusus untuk absen yang disediakan. Ketika karyawan memilih absen masuk ataupun absen pulang sistem secara otomatis akan mencatat jam absen tersebut dan langsung menyimpannya ke SAP pada saat itu juga.

b. Request Change for Personal Profile (Pengajuan Perubahan Data Pribadi)

Karyawan yang hendak mengajukan pengubahan data pribadi dapat menampilkan data yang hendak diubah tersebut terlebih dahulu dan kemudian memilih tombol *Request to Change*. Tipe data pribadi yang dapat diubah antara lain *Personal Data*, *Addresses*, *Family Member*, *Education*, dan *Personal ID*. Setelah karyawan memasukkan atau mengganti datanya yang lama dengan yang baru maka ia dapat melanjutkan ke proses selanjutnya yaitu dengan memilih tombol *Submit*. Kemudian pengubahan data tersebut akan disimpan sementara ke dalam SQL Server sambil menunggu persetujuan dari seorang approver yang berwenang dan dari pihak *Talent Management*. Approver dan pihak *Talent Management* dapat menampilkan data lama apa saja dari karyawan yang hendak diubah dengan data baru untuk kemudian disetujui pengubahan tersebut maka *update* data tersebut akan secara otomatis disimpan ke SAP pada saat itu juga.

c. Request for Employee Leave (Pengajuan Cuti)

Pada saat karyawan hendak mengajukan cuti, ia dapat membuka layar *Request for Employee Leave* dan mengisi seluruh formnya secara lengkap mulai dari jenis cuti, alasan cuti, alamat dan nomor telepon yang dapat dihubungi selama cuti, tanggal masuk kembali dan lain sebagainya untuk kemudian di-*submit* pada sistem dan disimpan sementara ke dalam SQL Server. Permohonan cuti tersebut akan dipilah berdasarkan selang waktu antara tanggal pengajuan dan tanggal cuti, apabila permohonan diajukan lebih dari atau sama dengan tujuh hari sebelum tanggal cuti maka permohonan tersebut hanya memerlukan sebuah persetujuan dari seorang *approver* namun apabila ternyata permohonan tersebut diajukan kurang dari tujuh hari dari tanggal cuti maka permohonan tersebut memerlukan dua buah persetujuan dari *approver* yang berwenang.

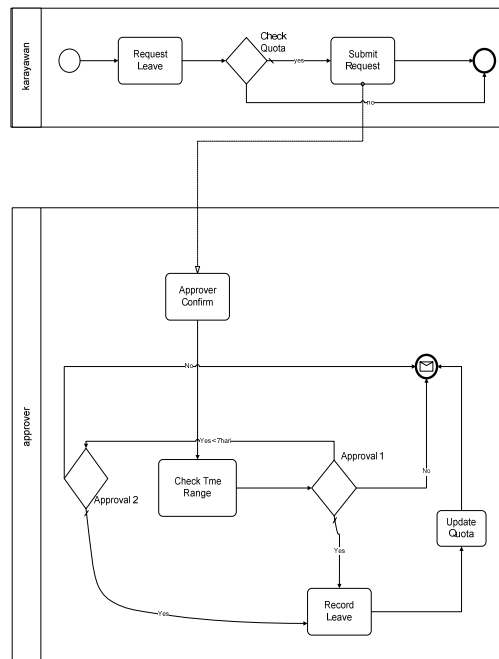
Ketika *approver* berwenang menerima permohonan cuti tersebut dan telah memberikan keputusan maka status permohonan cuti yang disimpan sementara dalam SQL Server akan berubah. Jika disetujui maka sistem akan mengecek permohonan cutinya apakah hanya membutuhkan sebuah persetujuan atau lebih dari itu, jika cuti ternyata membutuhkan sebuah persetujuan lagi maka permohonan tersebut akan disampaikan kepada *approver* berwenang berikutnya untuk diputuskan. Apabila permohonan cuti telah memperoleh seluruh persetujuan yang diperlukan maka cuti tersebut yang semula masih tersimpan sementara di dalam SQL Server akan secara otomatis disimpan ke SAP saat itu juga. Jika permohonan cuti tidak disetujui maka sistem pun tidak akan melakukan pemrosesan selanjutnya namun data tetap tersimpan.

d. Request for Employee Overtime (Pengajuan Lembur)

Pada saat karyawan hendak mengajukan lembur, ia dapat membuka layar *Request for Employee Overtime* pada HRIS dan mengisi form secara lengkap baik tanggal dan jam lembur untuk kemudian dimasukkan ke dalam sistem dan disimpan sementara di SQL Server kemudian permohonan lembur tersebut akan dipilah ke dalam dua golongan.

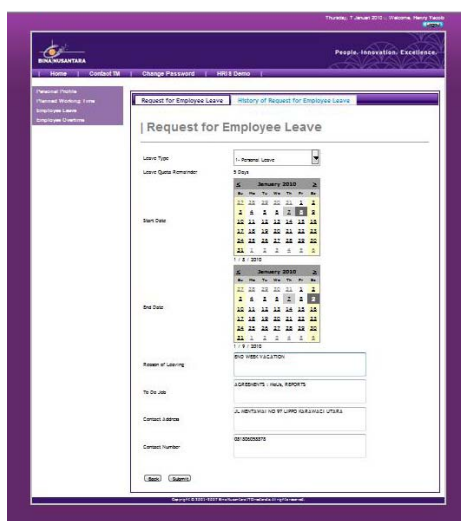
Apabila lembur yang diajukan kurang dari 10 jam maka permohonan tersebut hanya membutuhkan sebuah persetujuan dari *approver* yang berwenang dan persetujuan dari bagian TM, namun apabila lembur yang diajukan tersebut ternyata lebih dari atau sama dengan 10 jam maka permohonan tersebut akan membutuhkan dua persetujuan dari dua orang *approver* yang berwenang dan sebuah persetujuan bagian TM. Permohonan lembur dan telah diajukan tadi akan diterima oleh *approver* berwenang yang akan memberi keputusan, setiap keputusan yang diberikan akan mengubah status permohonan lembur yang tersimpan di SQL Server. Apabila *approver* tersebut telah menyetujuinya maka sistem akan mengecek apakah lembur tersebut masih memerlukan persetujuan dari *approver* yang lain atau tidak, jika perlu maka sistem pun akan menyampaikan permohonan lembur tersebut ke *approver* yang lain untuk diberi keputusan. Setelah mendapat seluruh persetujuan dari *approver* yang diperlukan maka permohonan lembur tadi akan disampaikan ke bagian TM untuk

diputuskan dan setelah TM pun menyetujuinya maka permohonan lembur pun yang semula berada di SQL Server akan disimpan dalam database SAP saat itu juga. Apabila permohonan lembur ditolak maka tidak ada pemrosesan selanjutnya, hanya saja data tetap tersimpan ke SAP. Semua aliran data yang berlangsung selama pengaksesan database SAP ini akan berjalan secara otomatis pada saat yang bersangkutan atau *real time* karena akses data pada sistem yang diusulkan ini memanfaatkan fasilitas *service* yang telah dibangun sebagai konektor guna perantara bagi HRIS dan SAP. Perubahan proses ini digambarkan dalam bentuk BPMN (*Business Process Medeling Notation*). Berikut contoh BPMN untuk proses *Profile Update*.

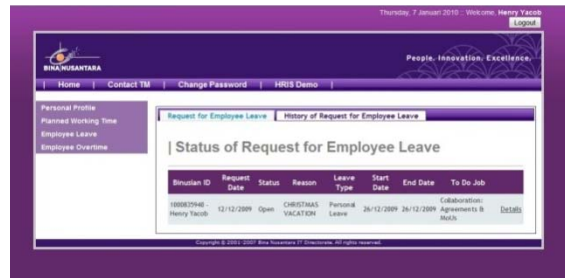


Gambar 8. BPMN Request for Employee Leave

UI (User Interface) untuk *Request for Employee Leave* dapat dilihat pada gambar berikut. UI ini digunakan untuk mengajukan cuti (*leave*) bagi karyawan.



Gambar 9. User Interface Request for Employee Leave

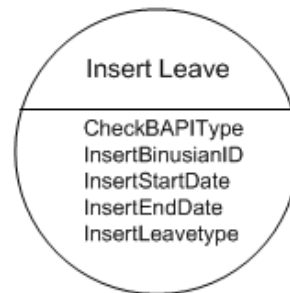


Gambar 10. User Interface History of Request for Employee Leave

Untuk service dan BAPI yang digunakan pada proses *Request for Employee Leave* dapat dilihat pada gambar berikut.



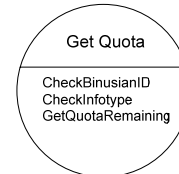
Gambar 11. Service *Get Leave Request Details*



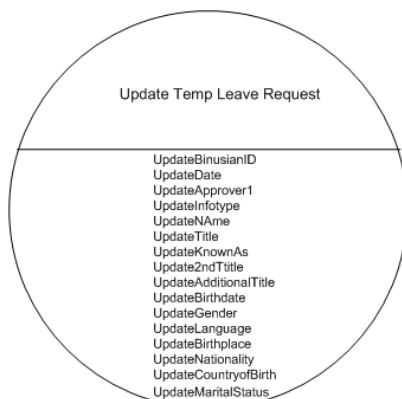
Gambar 12. Service *Insert Leave*



Gambar 13. Service *Leave Request Validation*



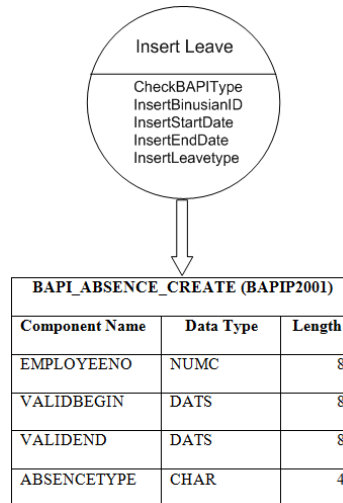
Gambar 14. Service *Get Quota*



Gambar 15. Service *Update Temp Leave Request*



Gambar 16. Service *Get Leave Request History*



Gambar 17. Service dan BAPI Employee Leave Details

SIMPULAN

Dari penelitian ini terlihat bahwa integrasi HRIS dengan SAP dengan menggunakan *services* berbasis *web* melalui analisis SOA yang memungkinkan pengubahan proses data dari *batch processing* menjadi *online processing* sehingga data dapat diakses secara *real time*. Selain itu *service* yang dibangun juga bersifat *reusable* karena dapat digunakan kembali untuk memudahkan pengembangan sistem aplikasi selanjutnya.

DAFTAR PUSTAKA

- BPMI.org (2004). Business Process Management Notation (BPMN).
- Cardoso, Jorge (2007). Semantic Web Services : Theory, Tools and Application. Information Sciences References, Portugal.
- Dabhi, Vipul K., Prajapati, Harshad B., Doshi, Varshil and Chokshi, Kadam. (2009). Developing Enterprise Solution with Web Services Integration. *International Journal of Web Services Practices*, Vol. 4, No.1, pp. 11-17
- Erl, Thomas (2005). Service-Oriented Architecture; Concepts; Technology; and Design, Prentice Hall PTR.
- Hernandes, Jose A., Keogh, Jim, and Martinez, Franklin Foster (2005). SAP R/3 Handbook. Third Edition. McGraw-Hill, USA
- Jarir, Z., Quafafou, M. and Erradi, M. (2010). Personalized Web Services for Web Information Extraction. *International Journal of Web Services Practices*, Vol. 5, No.1, pp. 22-31
- Meyliana, Yacob, Henry, Seymour, Onyx L., dan Crhristian (2010). Analisa dan Perancangan Services yang Menghubungkan HRIS dengan Modul HR SAP pada Bina Nusantara. Skripsi Tidak

Diterbitkan, Jakarta, BINUS University.

Signorile, Daniel Larocca (2007). SAP Query Reporting. Pearson Education, Inc by Sam Publishing, USA.

Tatnall, Arthur (2005). Web Portals : The New Gateways to Internet Information and Services. Idea Group Publishing, Australia